

Полтавцев А.А., Карельская К.А. Использование суррогатных ключей для идентификации объектов в реляционной базе данных. // Проблемы информатики в образовании, управлении, экономике и технике: Сб. статей XVII Междунар. научно-техн. конф. – Пенза: ПДЗ, 2017. – С. 136-139.

УДК 681.324(03)

## ИСПОЛЬЗОВАНИЕ СУРРОГАТНЫХ КЛЮЧЕЙ ДЛЯ ИДЕНТИФИКАЦИИ ОБЪЕКТОВ В РЕЛЯЦИОННОЙ БАЗЕ ДАННЫХ

А.А. Полтавцев, К.А. Карельская

## USING SURROGATE KEYS FOR OBJECTS IDENTIFICATION IN A RELATIONAL DATABASE

A.A. Poltavtsev, K.A. Karelskaya

**Аннотация.** В статье рассматриваются проблемы преобразования объектных данных в реляционные и обратно. Рассматривается одна из составляющих таких преобразований – взаимная уникальная идентификация объекта в бизнес-логике и базе данных. Анализируются варианты и обосновывается использование технологии суррогатных ключей.

**Ключевые слова:** базы данных, реляционная модель данных, объектная модель данных, преобразование моделей.

**Abstract.** The article deals with the problems of converting object data into relational ones and vice versa. One of the components of such transformations is considered - mutual unique of the object identification in the business logic and database. The options are analyzed and the use of surrogate key technology is justified.

**Keywords:** databases, relational data model, relational data model, models map.

До настоящего времени так и не решена проблема "импеданса рассогласования" между объектным и реляционным подходами. Условием успешного хранения в реляционных отношениях объектов является понимание обоих парадигм и их различий с тем, чтобы на основе этого понимания найти разумные компромиссы.

Объектная парадигма основывается на построении приложений из объектов, имеющих данные и поведение, реляционная парадигма основывается на хранении только данных. При объектном подходе мы перебираем объекты, используя взаимосвязи между ними, при реляционном подходе мы дублируем часть данных для объединения строк из различных отношений.

Поэтому необходима жизнеспособная стратегия эффективного моделирования потребностей по хранению объектов объектно-ориентированных приложений, организованная в некоторый "шаблон проектирования" (design pattern), т.е. необходим Шаблон Моделирования Хранения Объектов (**Persistence Modeling Process Pattern**). Это позволит создавать не только объектно-ориентированное приложение, но и модель данных на основе единой объектной модели. В свою очередь, построение и приложения, и механизма хранения на основе одной модели повышает шансы их эффективной совместной работы.

Тот факт, что реляционные модели данных являются неэффективным базисом для создания объектных моделей, обусловлен несколькими причинами. Одной из таких причин является то, что такие постулаты классической реляционной мо-

дели, как "ключевые атрибуты должны иметь прикладной смысл" или "применимость композитных ключей для идентификации сущностей", не применимы в объектной.

Для идентификации объектов необходимо иметь для них уникальные идентификаторы: по реляционной терминологии **PK** – первичные ключи сущности; **OID** – по объектной. В объектном приложении **OID** реализуется через создание специального класса. В объектном приложении **OID** может реализовываться программистом, если необходим обмен объектами в распределенных вычислениях (CORBA и др.), но может и не реализовываться, если приложение работает автономно и объект уникально определяется типом (именем класса) и именем объекта. В последнем случае **OID** все равно создается, но средой исполнения.

Наличие в реляционной базе данных (РБД) **OID** у объектов приносит определенные преимущества:

хотя в РБД нет необходимости осуществлять навигацию, тем не менее, необходимо осуществлять объединение таблиц, считывать агрегаты объектов. Наличие **OID** упрощает данные действия;

наличие **OID** упрощает также автоматизацию управления и восстановления взаимосвязей между объектами. Когда все таблицы с объектами имеют в качестве ключа поле одного и того же типа (**OID**), становится легко написать родовой код, использующий наличие данного факта.

В реляционной парадигме предполагается, что **PK** составляют одно или несколько полей, имеющих прикладной смысл. Однако для **OID** такой подход неприемлем. Дело в том, что бизнес-поля имеют свойство изменяться и, соответственно, может возникнуть необходимость, например, увеличить длину поля или изменить домен, на котором данное поле определено. Если данное поле используется еще и как первичный ключ, то данные изменения необходимо произвести во всех таблицах, где оно использовано как внешний ключ (**FK**). Такое действие, прежде всего, означает остановку и переделку объектного программного обеспечения, работающего на данной базе данных. Кроме того, при большом количестве ссылок, это может привести к большой работе и быть потенциальным источником ошибок. Так как единственной задачей **OID** в РБД является осуществление ассоциации **PK-FK**, для реализации и сохранения в БД связей между объектами, то такой подход для них неприемлем. Таким образом, *идентификатор объектов в РБД должен реализовываться как суррогатный ключ.*

На реализацию **OID** в базе данных как суррогатного ключа накладываются ограничения два основных соображения:

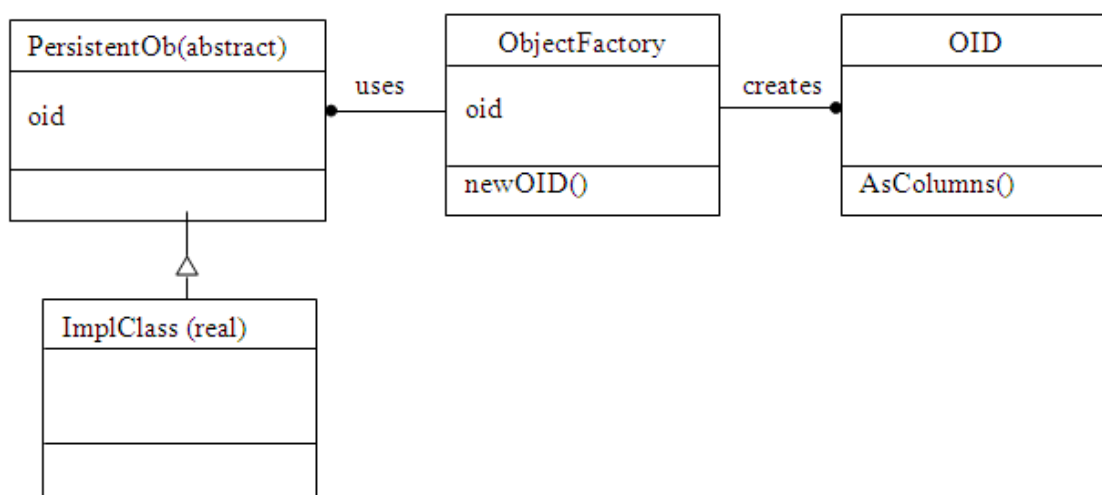
уровень уникальности **OID**: уникальность идентификатора объектов в пределах класса, уникальность в пределах всего дерева классов, уникальность в пределах всех объектов всех классов всех баз данных. Например, пусть класс **B** является наследником класса **A**, и уникальность реализована только в пределах класса. Тогда при запросе на выборку всех объектов типа **A** получим два различных объекта с одинаковыми **OID**. Поэтому минимальным уровнем уникальности является уникальность в пределах дерева иерархии классов;

алгоритм, который используется для формирования **OID**. Это может оказать значительное влияние на скорость работы с БД. Конечно, такой алгоритм может зависеть от того, какой уровень уникальности реализуется и какая стратегия отображения классов в таблицы принята.

Можно предложить различные алгоритмы построения **OID**. Например, можно использовать целочисленный столбец и функцию **MAX()** БД. Недостатками данного подхода являются необходимость блокирования всей таблицы, для вычисления **OID**, неиспользование идентификаторов объектов, удаленных из базы данных, а также возможность обеспечения уникальности **OID** только в пределах класса.

Можно использовать специальную таблицу в БД. В этом подходе возможны два варианта. Во-первых, можно использовать таблицу с одной строкой, содержащей последний **OID**. Данное значение увеличивается, и полученный результат присваивается объекту. Недостатком варианта является то, что такая таблица становится узким местом разработки, а содержащееся в ней значение крайне быстро возрастает.

Предлагаемая в работе стратегия состоит в построении проектного шаблона (рисунок) для класса реализации **OID** и отражения данного класса в таблицы РБД в соответствии с принятой стратегией отражения.



*Построение проектного шаблона*

Во-вторых, можно использовать двухстолбцовую таблицу, содержащую по строке для каждой из таблиц, содержащихся в БД. Первый столбец содержит имя таблицы в БД, а второй – текущее значение для первичного ключа данной таблицы. Как и в предыдущем варианте, эта таблица становится узким местом разработки, кроме того, обеспечивая первичные ключи для таблиц, она может не обеспечить уникальность **OID** для объектов, отображаемых в несколько таблиц.

В системе существует единственный экземпляр **ObjectFactory**, который и вычисляет новое значение, как только в системе создается новый хранимый объект. **ObjectFactory** создает экземпляр класса **OID**, который используется приложением как уникальный **OID** хранимого объекта. Метод **asColumns** возвращает коллекцию данных, которые должны быть сохранены в БД как экземпляр **OID**.

**Полтавцев Анатолий Алексеевич**  
Тверской государственный  
технический университет,  
г. Тверь, Россия  
E-mail: aapolt@gmail.com

**Poltavtsev A.A.**  
Tver State Technical University,  
Tver, Russia

**Карельская Катерина Александровна**  
Тверской государственный  
технический университет,  
г. Тверь, Россия  
E-mail: kak69@yandex.ru

**Karelskaya K.A.**  
Tver State Technical University  
Tver, Russia